

CLAIM AMENDMENTS

Pursuant to 37 CFR §121(c), the claim listing, including the text of the claims, will serve to replace all prior versions of the claims, in the application.

Claims 3 and 5-11 are pending. Claims 1, 2, 4 and 12-22 have been canceled, claim 3 has been amended to include the allowable features of claim 4, and claim 5 has been amended to depend from claim 3.

1. - 2. (Canceled)

3. (Currently Amended) A method for commonly controlling device drivers, comprising the steps of:

arranging a device independent access hierarchy between an application hierarchy and a device driver hierarchy;

defining functions available in a corresponding device driver among functions of a function block in a function table;

when a device is initialized, allowing said device independent access hierarchy to generate a device handler identifier having a standardized common data format for said device and transmit the generated device handler identifier having the standardized common data format to the application hierarchy of a higher order, with said device handler identifier being represented as DCB handlerId, where x1, x2 or x3 is an unsigned integer, x1 being a value of the level 1 meaning a device ID, x2 being a value of the level 2 meaning a logical or physical group number of a

13 corresponding device, x3 being a value of a channel meaning a channel number of a corresponding
14 device or group; and

15 allowing the higher-order application hierarchy to call a predetermined device using the
16 device handler identifier having the standardized common data format, and allowing said device
17 independent access hierarchy to identify a function of the corresponding device driver from the
18 function table using the device handler identifier having the standardized common data format and
19 call the function of the corresponding device driver.

1 4. (Canceled)

1 5. (Currently Amended) The method as set forth in claim [[4]] 3, with values of x1, x2 and
2 x3 being "0" corresponding to there being no corresponding level or channel and the value of x1
3 sequentially increasing from "1" when the device is initialized.

1 6. (Previously Presented) A method for commonly controlling device drivers, comprising
2 the steps of:

3 arranging a device independent access hierarchy between an application hierarchy and a
4 device driver hierarchy;

5 when a device initialization is controlled by said application hierarchy, allowing said device
6 independent access hierarchy to carry out level 1 initialization, level 2 initialization and channel
7 initialization and generate a device handler identifier having a standardized data format for devices;

8 allowing said device independent access hierarchy to dynamically assign a device control

9 block, containing elements for carrying out a standardized rule, corresponding to said device handler
10 identifier having the standardized data format;

11 allowing said device independent access hierarchy to provide said device handler identifier
12 to said application hierarchy; and

13 allowing said application hierarchy to call a predetermined device through said device
14 independent access hierarchy using said device handler identifier.

1 7. (Original) The method as set forth in claim 6, with the elements of said device control
2 block comprising a pointer of “*pControlTable” for pointing a position of a command control table,
3 the command control table containing a command identifier having a standardized unique value and
4 a command function pointer mapped to the command identifier, a pointer of “*pDDCB” for pointing
5 a position of a device driver control table through which the existence and position of a
6 corresponding function is identified, and a pointer “*pAnchor” for pointing a next level.

1 8. (Original) The method as set forth in claim 6, with the elements of said device control
2 block comprising a pointer of “*pHandler” for pointing a position of a given initialization profile
3 when a device is initialized, a function pointer of “*fpInitDevice” being used when a device is
4 initialized, a function pointer of “*fpOpenChannel” being used when a channel is open, a function
5 pointer of “*fpCloseChannel” being used when a channel is closed, a function pointer of “*fpRead”
6 being used when data of an open channel is read, a function pointer of “*fpWrite” being used when
7 data of the open channel is written, a function pointer of “*fpReset” being used when a device is
8 reset, a pointer of “*pControlTable” for pointing a position of a command control table containing

9 a command identifier having a standardized unique value and a command function pointer mapped
10 to the command identifier, a pointer of “*pDDCB” for pointing a position of a device driver control
11 table through which the existence and position of a corresponding function is identified, a pointer
12 of “*pEventTable” for pointing a position of an event table, and a pointer “*pAnchor” for pointing
13 a next level.

1 9. (Original) The method as set forth in claim 6, with the level 1 initialization of said device
2 being made by giving a device identifier value of x1 as a unique value for each device based on a
3 sequence of the level 1 initialization in the device handler identifier represented as DCB handlerId
4 where x1, x2 or x3 is an unsigned integer.

1 10. (Original) The method as set forth in claim 9, with the level 2 initialization of the device
2 being made by referring to the number of logical or physical groups, assigning anchors, and giving
3 a group value of x2 as a unique value for each anchor in the device handler identifier represented as
4 DCB handlerId where x1, x2 or x3 is an unsigned integer.

1 11. (Original) The method as set forth in claim 10, with the level 3 initialization of the
2 device being made by giving a channel value of x3 for each of channels belonging to the device and
3 groups within the device on the basis of an open channel sequence in the device handler identifier
4 represented as DCB handlerId where x1, x2 or x3 is an unsigned integer.

1 12.- 22. (Canceled)